Berechnung von Reihen und Konvergenzverhalten

Beispiel einer Reihenentwicklung mit $\sin(x)$

Als einfaches Beispiel für eine unendliche Reihe wird hier die Reihendarstellung der Funktion $\sin(x)$ verwendet. Diese ist gegeben durch

$$\begin{aligned} \sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots && (1) \\ &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \,, && (2) \end{aligned}$$

welche für alle $|x| < \infty$ konvergiert.

In allen numerischen Programmiersprachen kann man solche unendlichen Reihen natürlich nur näherungsweis auswerten, da die Summation von unendlich vielen Termen unendlich lange dauern würde. Daher gibt man sich mit der endlichen Teilsumme

$$\begin{aligned} \sin(x) &\approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} && (3) \\ &\approx \sum_{k=0}^{n} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \,, && (4) \end{aligned}$$

zufrieden. Abhängig von der gewählten Reihe und von den Werten für x, konvergieren sie langsamer, schneller oder eben gar nicht zum gewünschten Wert.

Reihenentwicklung in MATLAB

Summen dieser Art kann man in MATLAB sehr gut ohne die Verwendung von `for`-Schleifen programmieren. Nützlich sind dabei folgende Informationen:

- Äußerst praktisch für die Berechnung solcher Reihen ist der Befehl `meshgrid`. Damit kann man aus einem Vektor **x** mit allen x-Werten und einem Vektor **k** mit allen k-Werten zwei gleich große Matrizen

  ```
  x = [-2:2]; k = [0:5];
  [xx,kk] = meshgrid(x,k);
  ```

  erzeugen

  $$xx = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \qquad kk = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \,. \qquad (5)$$

  Mit dem Befehl `A=xx.^(2*kk+1)` kann man nun z.B. x^{2k+1} für alle Werte von x und k gleichzeitig ausrechnen.

  $$A = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -8 & -1 & 0 & 1 & 8 \\ -32 & -1 & 0 & 1 & 32 \\ -128 & -1 & 0 & 1 & 128 \\ -512 & -1 & 0 & 1 & 512 \\ -2048 & -1 & 0 & 1 & 2048 \end{bmatrix} \,, \qquad (6)$$

- Die Summe über alle k-Werte kann man dann mit Hilfe des `sum(A,1)`-Befehles erhalten:

```
S=sum(A,1)
S =
 -2730          -6            0            6         2730
```

Dabei wird jedes Element von S, $\{\mathbf{S}\}_\mathbf{x}$ durch Aufsumieren über die x-te Spalte berechnet

$$\{\mathbf{S}\}_\mathbf{x} = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}} \mathbf{A}_{\mathbf{k},\mathbf{x}} = \mathbf{A}_{\mathbf{1},\mathbf{x}} + \cdots + \mathbf{A}_{\mathbf{n},\mathbf{x}}.$$

Da MATLABauf Rechnen mit Matrizen optimiert ist, läuft das Programm bei größeren Datenmengen so wesentlich schneller als mit den sonst üblichen `for` Schleifen.

- Statt mit der Summe kann die gesamte Reihenauswertung auch mit Hilfe der Matrizenmultiplikation durchgeführt werden. Aus der linearen Algebra ist bekannt, dass für einen $(1 \times n)$-Vektor $\mathbf{f}$ (Zeilenvektor) und eine $(n \times m)$-Matrix $\mathbf{A}$ gilt

$$\{\mathbf{fA}\}_\mathbf{l} = \sum_\mathbf{k} \mathbf{f_k A_{kl}} \ . \tag{7}$$

Erzeugt man nun für alle Werte von k den Vektor $\mathbf{f} = (-\mathbf{1})^\mathbf{k}/(\mathbf{2k}+\mathbf{1})!$, kann man mit $\mathbf{f} \cdot \mathbf{A}$ (in MATLAB `f*A`) für alle Werte von x gleichzeitig den jeweiligen Wert der Teilsumme erhalten. (Mit diesen speziellen Beispielen für $\mathbf{A}$ und $\mathbf{f}$ bekomt man die Reihenentwicklung für den Sinus.)